REMARKS

Applicants thank the Examiners for the time spent with applicants' attorney during the recent Interview. Reconsideration and allowance for the above-identified application is respectfully requested in view of the amendments and remarks herein.

Independent claims 1 and 19, directed to the claimed method and a corresponding computer program product, have been canceled and replaced by new claims 32 and 33, respectively. Claims 5, 7, 8, 14, 22 – 24, 30 and 31 have been canceled. Thus, the claims presented for reconsideration are independent method claim 32, with depending claims 2 – 4, 6, 9 – 13 and 15; independent method claim 16, with depending claims 17 – 18, and computer program product claim 33 (corresponding to the method of claim 1), with depending claims 20, 21 and 25 – 29. Minor amendments have been made to the specification, as noted, to correct minor errors.

At the Interview an extended discussion was had concerning what appeared to the Examiner to be a proper basis by which the applicants' claimed system would be patentably distinguished over the prior art. In particular, the Examiner noted in the Interview Summary that amending the claims to reflect that there is a distinction between the database used by the client applications and implementation of the notification system at the level of the operating system, that this would overcome the prior art of record.

Independent claim 32 and its computer program counterpart, claim 33, have taken the Examiner's comment into account.[1] In particular, in the preamble to these claims the nature of the architecture controlled by the claimed method is set forth as

[1] These claims are clearly supported by Figures 4 and 5, and the accompanying disclosure at pages 15 – 18.

> "A computing system comprised of an *operating system level* and *client applications*, and that includes,
>
> *at the level of the operating system*, a database engine for receiving the high-level commands from the database application and then implementing them by executing object commands passed to the operating system, the high level commands being issued to the database engine from a database application that generates high-level document commands relating to operations to be performed on a document such as a folder, a file, a message or other such entities that are identified at the level of client applications, and
>
> *at the level of the client applications*, a database containing one or more tables each comprised of objects that define various properties of one or more documents, and wherein the objects contained in the tables of the database are updated in response to the object commands received from the operating system." Emphasis added.

The claim then goes on to define a "method for *asynchronously* notifying client applications of the implementation of particular high-level document commands *so that such notification survives even failures of either the database engine or client applications*." Emphasis added.

As defined in the claims, the method of asynchronous notification is comprised of

> "[T]he database application issuing one or more high-level document commands which are to be implemented;
>
> either i) before implementing an issued high-level document command, or ii) at the same time that any other table of the database is updated, the database engine [*this is at the operating system level, as set forth in the preamble*] altering a persistently stored notification table *located at the operating system level,* by creating a notification entry to reflect the implementation of the issued high-level document command, with the result that the client applications do not affect the implementation of the high-level document command since the implementation takes place prior to notification of the client applications, even in the event of failure of either database or the client application, because either,
>
> > for case i), no tables in the database are updated until such failure is restored and notification of implementation is verified by at least one of the client applications to the database engine, or
> >
> > for case ii), all tables in the database are updated in the same transaction used to created the notification entry when updating the persistently stored notification table of the database engine;
>
> *dispatching from the operating system a notification* of the implementation of the issued high-level document command to one or more client applications requiring such notification;
>
> receiving acknowledgement from at least one of the one or more client applications that the notification has been received; and

> in response to receiving acknowledgement, the database engine [*on the operating system level*]altering the notification table to reflect that no client applications any longer need to be notified of the implementation of the high-level document command." Parenthetical statements and emphasis added.

Consistent with the discussion of these claims at the Interview, independent claims 32 and 33 reflect the distinction between i) the database used by the client applications, and ii) implementation of the asynchronous notification by updating (either prior to or concurrent with updating the tables of the database) the notification table at the level of the operating system. Thus, these claims and their dependent claims are patentable over the prior art of record for the reasons noted by the Examiner in the Interview Summary.

As for independent claim 16, the remaining independent claim at issue, in the Office Action that claim (as well as the other independent claims) was rejected under 35 U.S.C. § 103(a) as obvious over U. S. Pat. No. 5,504,897 to Gans et al. (hereinafter "Gans"), in view of the publication "COM+ Technical Series: Loosely coupled Events" (hereinafter "LCE"). Various of the dependent claims were also rejected as obvious over Gans in view of LCE and further in view of U. S. Pat. No. 6,018,762 to Brunson et al. (hereinafter "Brunson").[2]

As noted during the Interview Gans discloses a method which is intended to provide the ability to scale an electronic mail system. Col. 1 lines 53 – 55. Scalability is accomplished by processing mail entries in parallel to accommodate increased messaging activity. Id. at lines 55 – 56. This is in turn accomplished by assigning a message to a particular queue, rather than to a particular process. Then multiple processes can be assigned to process a queue. Thus, as more activity causes the number of entries in a queue to increase, additional processes can be assigned

[2] Brunson qualifies as "prior art," if at all, under 35 U.S.C. § 102(e). Any reference or argument herein to Brunson is not be construed as an acceptance that Brunson is a proper qualifying reference, and applicants thus specifically reserve the right to challenge the "prior" art status of Brunson at any time. Moreover, applicants do not necessarily acquiesce in the merits of any rejection asserted by the Examiner and the absence herein of any reply, express or otherwise, to any such rejection should not be construed as acquiescence to such.

to process the queue's entries. Id. at lines 62 – 67. Gans incidentally discloses that "Any method can be used to identify queue entries previously or currently being processed by one process. In the preferred embodiment, messaging and process information are stored in a relational database system that provides the ability to perform locking at the record level." Col 2 lines 14 – 18. Using the relational database with record locking capability, as each queue entry (e.g., row in the database) is selected for processing it can be locked so that it cannot be processed by any other process. Id. at lines 49 – 57.

It is clear from the foregoing that Gans is not specifically concerned with, nor does Gans in any way address the problem solved by applicants' claimed invention, namely, how to provide "asynchronous" notification, meaning that client applications do not affect the implementation of the high-level document command since the implementation takes place prior to the notification. Gans provides no disclosed solution, express or suggested, to the possibility that the high-level document command is implemented in the database, but that the subscribing client application never receives notification of the implementation due to system failures of one type or another.

The Office Action acknowledges (see p. 3) that Gans "does not explicitly disclose a method for reliably notifying client applications of the implementation of a particular high-level document command in the database." Reliance is thus placed on LCE to show that it would have been obvious to a person of ordinary skill in the art at the time of applicants' invention to modify Gans by using LCE.

The asserted combination of Gans, as modified by LCE is improper and fails to rise to the level of a *prima facie* case in the first instance because there is no showing in the Office Action as to why one of skill in the art, based on Gans teachings in the first instance, would be motivated to combine the LCE teachings with Gans' teaching.[3] For that reason alone, the rejection should be reconsidered and withdrawn even in the absence of any amendment to claim 16.

As noted above, there is no appreciation in Gans or the other prior art of the problem addressed by applicants' invention.[4] Indeed, in that sense, Gans actually teaches away from such motivation. The Office Action notes (p. 3) that "Gans'879 discloses use of the Oracle ™ relational database which is well known to have a database level transaction monitor." That being the case, it seems manifestly illogical that one of skill in the art would seek to replace or modify the transaction monitor of Gans' disclosed Oracle™ relational database with another one, particularly with the LCE technology of a well-known competitor whose technology is not compatible with that of Oracle's.[5]

Even were the requisite motivation to exist within the prior art, which is not the case here, the asserted combination in any event does not meet the required claim limitations of amended claim 16, and for that additional reason, fails to make the required *prima facie* case to sustain any continuing rejection of claim 16, as noted further below.[6]

In particular, claim 16 as defines a method for use in "a database management system that includes a database engine that accesses and updates objects in a database, the database

[3] A determination of obviousness must involve more than indiscriminately combining prior art. A motivation or suggestion to combine the references must exist. *Micro Chem., Inc. v. Great Plains Chem. Co.*, 103F.3d 1538 (Fed. Cir. 1983). That suggestion or motivation must be found in the prior art, as whole, which must somehow suggest the desirability of making such a combination of the teachings found in the art. *Akzo N.V. v. Unnited States ITC*, 808 F.2d 1471(Fed. Cir. 1986). The absence of a suggestion to combine is dispositive in an obviousness determination. *Gambro Lundia AB v. Baxter Healthcare Corp.*, 110 F.3d 1573 (Fed. Cir. 1997).

[4] The problem confronted by the inventor must be considered in determining whether it would have been obvious to combine references in order to solve that problem. *Northern Telecom, Inc. v. Datapoint Corp.*, 908 F.2d 931(Fed. Cir. 1990).

[5] Both the suggestion *and the expectation of success* that the asserted combination can be made and will work, must be founded in the prior art, not in the applicant's disclosure. *In re O'Farrel*, 853 F.2d 894 (Fed. Cir. 1988).

[6] The questions to be resolved are first, whether sufficient motivation to make the asserted combination exists within the prior art, and second, whether, in any event, the invention as claimed achieves more than a combination which any or all of the references suggest. *Smithlkine Diagnostices, Inc. v. Helena Labs. Corp.*, 859 F.2d 878 (Fed. Cir. 1988).

engine receiving high-level document commands, each high-level document command for performing an operation on a document that is associated with a plurality of tables in the database." As claimed, claim 16 defines "a method for reliably notifying client applications of the implementation of particular high-level document commands in the database, *the notification surviving even system failures*." Emphasis added.

The method is performed by implementing a high-level document command in the database, and then "*ensuring asynchronous notification* of the high-level document command's implementation *by preserving the high-level command until all of the client applications to be notified acknowledge at least receipt of the notification, thereby ensuring that the client applications to be notified do not affect the implementation of the high-level document command since the implementation takes place prior to such notification*." Emphasis added.

As claimed, applicants' method of asynchronous notification is not found anywhere in the prior art of record, including the references discussed at the Interview. Accordingly, for at least the foregoing reasons, claim 16, like claims 32 and 33, is patentable over the prior art of record and therefore favorable reconsideration is requested.

In the event the Examiner finds any remaining impediment to allowance of this application that may be clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney.

Dated this 24th day of June, 2004.

Respectfully submitted,

Rick D. Nydegger

RICK D. NYDEGGER
Registration No. 28,651

WORKMAN NYDEGGER
Attorneys for Applicant
Telephone: (801) 533-9800
Customer No. 022913

RDN:dmh
DMS0000006430V001